IMPROVED LIGHTING AND SHADOWING METHODS
AND ARRANGEMENTS FOR USE IN COMPUTER GRAPHIC SIMULATIONS

Technical Field of the Invention

The present invention relates to computer graphics and,
more particularly, to improved methods and arrangements for
use in rendering lighting and shadows in computer graphic
5   simulations, such as, for example, interactive computer
graphics simulations of multi-dimensional objects.

Background

Computer generated graphics are becoming more popular in
everyday computing especially given recent advances in
10   affordable computer hardware and software. This trend is
expected to continue in the foreseeable future. By way of

example, interactive computer generated graphical images are becoming increasingly detailed and more realistic due to increased processing capabilities in personal computers (PCs). No where is this more evident than in the computer gaming

5 arena, wherein virtual three-dimensional (3D) worlds are created in software and interactively explored by the computer's user.

As with the real World, these virtual 3D worlds consist of a plurality of 3D objects. These 3D objects are typically

10 modeled by one or more polygons. In the computer, depending upon the orientation of the scene being viewed, some of these polygons correspond to individual pixels that are displayed on the computer's display screen. Within the computer, these pixels are represented by data that can be manipulated and

15 stored in at least one data base. For example, once the arrangement of these 3D objects with respect to one another has been determined for a particular viewpoint, the rendered scene of the 3D world is projected onto a two-dimensional (2D) frame that can be displayed to the user. The frame is

20 typically a data base that contains pixel color information with regard to the display screen.

A sense of motion within this virtual 3D world is provided by incrementally altering the arrangement of these 3D

2

objects and quickly re-displaying the transformed frame. Typically, a frame rate of at least about twenty-five (25) frames-per-second (fps) is desired to convey a sense of motion.

5   Rendering such interactive 3D worlds, however, typically requires that millions of calculations be conducted between frames (i.e., in "real time"). Since there is a limit to the amount of processing that a computer can provide between frames, simplifications or other compromises often need to be

10   made in modeling a 3D world. Additionally, advanced algorithms are implemented in either hardware and/or software to further streamline the processing of the resulting 3D scene and 2D images. Regardless of how the rendering computations are completed, the goal remains the same, namely, to provide

15   a realistic, interactive virtual 3D world to the user.

One of the unfortunate compromises made in the past, has been in the area of lighting and, more particularly, in the area of rendering shadows cast by lighted 3D objects. Many shadow rendering processes have been considered to be too

20   compute intensive for most lower-end computer applications, and as such shadow rendering is often ignored or otherwise greatly simplified. Several advanced and simplified shadow rendering algorithms and other graphical algorithms and

3

techniques are described by James D. Foley, et al. in *Computer Graphics: Principles and Practice*, second edition, 1997 (ISBN 0-201-84840-6), published by Addison-Wesley Publishing Company, Inc. This text is expressly incorporated herein, by

5 reference, in its entirety and for all purposes.

By way of a particular example, Foley et al. describe a promising two-pass object-precision shadow algorithm in Section 16.4.2 of the text. This two-pass shadow algorithm, developed by Atherton, Weiler and Greenberg, essentially

10 determines which portions on a 3D object's surfaces are visible from the perspective of a light source (i.e., from the light source's view point). This requires converting the data for the 3D object, as represented in a data base, from a modeled world coordinate to a light source coordinate and then

15 determining which portions of the various polygons are visible surfaces with respect to the light source, for example, using a hidden-surface removal algorithm. Since these visible portions are illuminated (i.e., lit) by the light source, the remaining portions (i.e., the hidden-surfaces) are darkened as

20 being shaded from the light. The resulting data is then transformed back to modeling world coordinates and merged with the original object data base. This creates a viewpoint-independent merged data base that can be used to determine the

4

shadows on the various 3D objects from any other viewpoint. This is the first step of the two-step shadow algorithm.

In the second step of the two-pass shadow algorithm, the data in the merged data base is then converted from the modeled world space to a corresponding screen (or camera) viewpoint. Then, a hidden-surface algorithm is used to determine which portions of the various polygons are visible surfaces with respect to the camera. These visible portions are identified as being visible to the camera and as being either not-darkened (i.e., lit) or darkened (i.e., in a shadow). A polygon scan-conversion algorithm is then used to render the resulting 2D image.

One of the drawbacks to this type of shadow algorithm is that unrealistic shadowing can occur in scenes having multiple light sources. Foley et al. for example, one page 747 of their text, teach that multiple light sources are handled by re-processing the merged data base for each new light source. Thus, for example, assume that a portion of a polygon which is visible from the camera viewpoint has been darkened because it is shaded from a first light source. This is information is recorded in the merged data base. Then, this merged data base is used as the starting point for a similar process for the second light source's viewpoint determination. As part of

5

this second light source's viewpoint determination, additional darkening can occur to portions of the scene that are also shaded from the second light source. As a result, there is a potential for a portion of the scene to be darkened twice. In

5   scenes having multiple light sources, including ambient light, this additive darkening of portions, and in particular, pixels, tends to reduce the realism of the shadowing effects and can lead to awkward shadowing cross-talk, wherein a shadow is made overly dark and has unrealistic transitions. This is

10   especially true for applications using a "stencil buffer" technique for shadowing.

Additionally, when there are several light sources, for example, four or more, the transformation process associated with converting between the camera, world, and light

15   viewpoints in support of this type of two pass algorithm can prove to be too burdensome, thereby inhibiting or otherwise diminishing the effectiveness of the resulting interactive real-time graphics.

Consequently, there is need for improved shadow rendering

20   methods and arrangements. Preferably, the improved shadow rendering methods and arrangements support real time interactive graphics on conventional PCs and the like, and

allow for multiple light sources to be modeled in a more efficient and realistic manner.

SUMMARY OF THE INVENTION

In accordance with certain aspects of the present invention, improved lighting and shadow rendering methods and arrangements are provided. These improved lighting and shadow rendering methods and arrangements can be implemented to support real time interactive graphics on conventional computers such as personal computers (PCs), computer gaming devices and the like. These improved methods and arrangements allow for multiple light sources to be modeled in a more efficient and realistic manner without becoming overly burdensome, or otherwise inhibiting or diminishing the effectiveness of the resulting interactive real time graphics.

For example, the above stated needs and others are met by a shadow rendering method for use in a computer system, in accordance with certain embodiments of the present invention. The method includes the steps of providing observer data of a simulated multi-dimensional scene and providing lighting data associated with a plurality of simulated light sources arranged to illuminate the scene. This lighting data includes light image data, for example. For each of the light sources,

the method then includes the steps of comparing at least a portion of the observer data with at least a portion of the lighting data to determine if a modeled point within the scene is illuminated by the light source, and storing at least a

5   portion of the light image data associated with the point and the light source in a light accumulation buffer. Once this is completed, the method includes the steps of combining at least a portion of the light accumulation buffer with the observer data, and displaying the resulting image data to a computer

10   screen.

In accordance with certain further embodiments of the present invention, the observer data includes observed color data and observed depth data associated with a plurality of modeled polygons within the scene as rendered from an

15   observer's perspective. Thus, for example, the modeled polygons can be associated with a single pixel on the computer screen or a group of pixels, wherein the observed color data includes an observed red-green-blue value for the pixel(s) and the observed depth data includes an observed z-buffer value

20   for the pixel(s). Similar data is provided for the lighting data, wherein, in certain embodiments, the lighting data includes source color data associated with at least one of the light sources and source depth data associated with the

8

plurality of modeled polygons within the scene as rendered from a plurality of different light source's perspectives.

In accordance with still further embodiments of the above method, at least a portion of the observed depth data is compared with at least a portion of the source depth data to determine if the modeled point is illuminated by the light source. This can include, for example, converting at least a portion of the observed depth data from the observer's perspective to at least one of the plurality of different light source's perspectives, before comparing the observed depth data with the source depth data. For fixed light sources and/or observer a precalculated matrix transformation look-up table can be used.

In accordance with other embodiments of the present invention, the above method can also be used to simulate dynamically changing light sources, interrupted light beams, reflected light beams, and/or projected light images, such as, for example, motion picture, video, animation, and computer graphics images. This can be accomplished, for example, by having at least a portion of the source color data being selectively controlled source color data, which can be changed over a period of time.

9

An arrangement that is configured to render shadows in a simulated multi-dimensional scene is also provided to meet the above stated needs and others, in accordance with certain further embodiments of the present invention. The arrangement

5   includes a display screen configured to display image data, memory for storing data, such as, observer data, lighting data, light accumulation data, and frame data. At least one processor is coupled to the memory and the display screen and operatively configured to compare, for each of the plurality

10   of light sources, observer data with lighting data to determine if a modeled point within the scene is illuminated by the light source and, if so, to store light image data in a light accumulation buffer, which can then be combined with observer data, and stored in the frame buffer for display via

15   the display screen.

In accordance with still further embodiments of the present invention a method for simulating light falling on a modeled object in a computer generated multi-dimensional graphics simulation is provided. This method includes

20   rendering a camera view of an object to produce a camera depth array and a corresponding camera image array, rendering at least one light view of the object to produce a light depth array and a corresponding light image array, and transforming

10

camera depth data to the light view to produce a transformed camera array. The method further includes, for each data value therein, comparing the light depth array to the transformed camera array to determine if the data value in the

5   light depth array is closer to the simulated light, and if so, adding a corresponding data value from the light image array to a light accumulation array, and then for each data value therein, multiplying the camera image array by a corresponding data value from the light accumulation array to produce a

10 lighted camera image array.

The improved methods and arrangements of the present invention can be implemented in hardware and/or software. Thus, in accordance with certain embodiments of the present invention, a computer-readable medium is provided. A computer

15 readable medium, by way of example, can include any tangible apparatus suitable for conveying or otherwise carrying at least one set of computer instructions. This can include a removable media associated with a computer system or a data communications link or network over which computer implemented

20 instructions and/or data are carried, for example.

Thus, in accordance with certain embodiments of the present invention, such computer instructions are configured to cause a computer to operatively simulate light falling on

11

a modeled object in a computer generated multi-dimensional graphics simulation by performing the following operations: 1) rendering an observer view; 2) rendering a source view; 3) transforming observed depth values to the source view;

5   4) modifying at least one image accumulation value with one of the observed image values if the corresponding transformed observer value is equal to a comparable one of the source depth values; 5) multiplying the one of the observed image values by the at least one image accumulation value to produce

10   at least one pixel value; and 6) displaying the pixel value on a computer screen. In certain embodiments, following operation 4), operations 2) through 4) are repeated for each additional source.

In accordance with yet other embodiments of the present

15   invention, there is provided a computer-readable medium carrying at least one set of computer instructions configured to cause at least one processor within a computer system to operatively render simulated shadows in a multi-dimensional simulated scene by performing the steps of providing observer

20   data, providing lighting data, comparing observer data with lighting data to determine if a modeled point within the scene is illuminated by the light source, storing at least a portion of the light image data associated with the point and the

12

light source in a light accumulation buffer, combining at least a portion of the light accumulation buffer with the observer data, and displaying resulting image data to a computer screen.

5

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the method and arrangements of the present invention may be had by reference to the following detailed description when taken in

10 conjunction with the accompanying drawings wherein:

Fig. 1 is a block diagram depicting a computer system that is configured to generate and display interactive 3D graphics, in accordance with certain embodiments of the present invention;

15 Fig. 2 is a block diagram depicting an exemplary 3D graphics scene that can be rendered by a computer system as in Fig. 1, in accordance with certain embodiments of the present invention;

Fig. 3 is a block diagram depicting exemplary

20 arrangements of modeling data as used to render shadows, for example, using the computer system in Fig. 1, in accordance with certain embodiments of the present invention;

13

Fig. 4 is an exemplary flow-chart depicting an improved shadow rendering process, for use in a computer system, for example, as in Fig. 1, in accordance with certain embodiments of the present invention;

5   Fig. 5 is an exemplary flow-chart depicting further enhancements to the improved shadow rendering process in Fig. 4, in accordance with certain further embodiments of the present invention;

Fig. 6 is an exemplary flow-chart depicting further enhancements to the process in Fig. 5, in accordance with still further embodiments of the present invention; and

10

Figs. 7A-D are photocopies of exemplary computer screen print-outs depicting an exemplary 3D scene and related rendered depth and image data from an interactive real time graphics rendering application, in accordance with certain embodiments of the present invention.

15

DETAILED DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram depicting a computer system or like device, 40, that is configured to generate and display interactive 3D graphics, in accordance with certain embodiments of the present invention. Computer system 40 includes at least one processor 42 that is configured to

20

14

operate in accordance with at least one computer application 44 having computer instructions therein. Processor 42 can include at least one microprocessor or central processing unit (CPU), math co-processors, input/output processors, memory

5   access processors, local memory cache, graphics accelerator processors, etc. A bus 46 is connected to processor 42 and configured to carry data and/or control signals thereto and therefrom. Bus 46 includes one or more buses, and can also include additional communication chip sets, interfaces,

10   gateways, and/or networks.

A primary memory 48, including at least one frame buffer 50, is connected to bus 46 and configured to store data and to be accessed by at least processor 42 via bus 46. Primary memory 48 can also include a plurality of additional memory

15   partitions, buffers, and/or data bases 51. Primary memory 48 typically includes random access memory (RAM) or the like. A secondary memory 52 is connected to bus 46 and configured to store data and to be accessed by at least processor 42 via bus 46. Secondary memory 52 can include additional memory and/or

20   a data storage device, such as, for example, a hard disk drive, removable-disk drive, a CD-ROM drive, a DVD-ROM drive, a tape drive, a server, and/or the like. A removable computer-readable media 53 is shown for use with certain types

15

of secondary memory 52. Media 53 can be, for example, an optical disc (e.g., CD, DVD, etc.), a magnetic disk, a data cartridge, a magnetic tape, or a similar medium configured to store data.

5   At least one user input device 54 is connected to bus 46 and configured to allow a user to input commands to processor 42 via bus 46. By way of example, input device 54 can include a keyboard, a mouse, a trackball, a game input, a joy stick, a pen and tablet, a pointing device, a touch screen, a voice

10   recognition device, or other similar devices. At least one output device 56 is connected to bus 46 and configured to display or otherwise output data from processor 42 to the user. For example, output device 56 can include a CRT display, a flat panel display, a LCD display, a plasma

15   display, a television, a projected display, a printer, an imager, etc.

Fig. 2 is a block diagram depicting an exemplary 3D graphics scene 10 that can be rendered by computer system 40 of Fig. 1. Scene 10 includes at least one 3D object 12 that

20   is arranged within a defined space (e.g., background or backdrop). 3D object 12 is represented by spacial data, for example, a plurality of polygons. When viewed from camera perspective 14, certain portions of scene 10 will be visible

16

within the field-of-view (fov) associated with the camera and the camera's position and orientation with respect to scene 10. Camera 14 is typically modeled as being responsive to visible light from scene 10. As such, scene 10 is

5   illuminated, for example, by one or more light sources, such as, a first light source 16 (#1) and a second light source 18 (#2). Light source #1, in this example, is a uni-directional light source having a particular position, orientation and constrained field-of-view. Light source #2 is, in this

10   example, a multi-directional light source having a particular position, orientation and non-constrained field-of-view. Light sources #1 and #2 are modeled as emitting one or more frequencies of visible light, at an initial intensity (color), onto scene 10.

15   A world space coordinate system 20, for example, having an arbitrary origin and x, y, and z coordinates extending therefrom, is employed to define the spacial relationship between 3D object 10, light sources 16 and 18, and camera 14. Known matrix algebra techniques are used to convert or

20   transform between world space coordinate system 20 and the unique coordinates from the perspectives of light source 16, light source 18, and camera 14. Such techniques are described, for example, in the text by Foley et al.

17

With this in mind, Fig. 3 is a block diagram depicting exemplary arrangements of modeling data as used to render lighting and shadows, in accordance with certain embodiments of the present invention. For convenience, the various

5   rendered data in Fig. 3 is illustrated as being logically stored in a plurality of 2D matrices or data bases 50 and 51A-G, each having an X axis and a Y axis. Preferably, and in this example, the X and Y axis correspond an exemplary output device 56 having a screen that displays X by Y number of

10   pixels when provided corresponding red-green-blue (RGB) pixel data values. Those skilled in the art will recognize that other conventions and/or arrangements can also be used for storing and manipulating the data.

For camera 14, a camera image 51A is rendered of scene

15   10. Camera image 51A includes RGB pixel data values from scene 10 for X by Y number of pixels (or pixel groups). An exemplary pixel 60 having camera coordinates SP_x and SP_y is depicted. Associated with camera image 51A is camera depth 51B. Camera depth 51B is a Z-buffer containing depth

20   information for each of the X by Y number of pixels in camera image 51A. The depth image is used during rendering to correctly determine the ordering of 3D objects within scene

18

10, from the camera's perspective. The use of Z-buffers is well known, for example, see Foley et al.

For light source #1, a light image 51C is rendered of scene 10. Light image 51C includes RGB pixel data values for the light emitted, for X by Y number of pixels. For example, the data in light image 51C can represent the intensity, color, and/or pattern of light emitted by light source #1. An exemplary pixel 60', which corresponds to pixel 60 in camera image 51A, is depicted following transformation as having light source #1 coordinates L_1P_x and L_1P_y. Associated with light image 51C is light depth 51D. Light depth 51D is a Z-buffer containing depth information for each of the X by Y number of pixels (or groups of pixels) in light image 51C. The depth image is used during rendering to correctly determine the ordering of 3D objects within scene 10, from light source #1's perspective.

Similarly, for light source #2, a light image 51E is rendered of scene 10. Light image 51E includes RGB pixel data values for the light emitted, for X by Y number of pixels (or groups of pixels). For example, the data in light image 51E can represent the intensity, color, and/or pattern of light emitted by light source #2. An exemplary pixel 60", which corresponds to pixel 60 in camera image 51A, is depicted

19

following transformation as having light source #2 coordinates L_2P_x and L_2P_y. Associated with light image 51E is light depth 51F. Light depth 51F is a Z-buffer containing depth information for each of the X by Y number of pixels in light image 51E. The depth image is used during rendering to correctly determine the ordering of 3D objects within scene 10, from light source #2's perspective.

A light accumulation buffer 51G is also depicted as having pixel data values for X by Y number of pixels (or groups of pixels), as in camera image 51A and camera depth 51B. Light accumulation buffer 51G includes RGB pixel data for the accumulated light falling on a pixel (or groups of pixels) from light sources #1 and #2. As discussed below, by using light accumulation buffer 51G, a more realistic shadow rendering can occur, because pixels are accumulatively lighted, rather than accumulatively darkened as in past interactive real time shadow rendering algorithms.

Frame buffer 50 is also depicted as having pixel data values for X by Y number of pixels, as in camera image 51A, camera depth 51B, and light accumulation buffer 51G. Frame buffer 50 includes RGB data of the final rendered 2D image of scene 10. The data in frame buffer 50 is provided to output device 56 for display to the user. Those skilled in the art

20

will recognize that a plurality of frame buffers can be employed to allow for additional buffering of frames.

A first transformation table (#1) 51H is also depicted in Fig. 3. Transformation table #1, is used, in accordance with certain embodiments of the invention to provide for rapid transformation of pixels between camera coordinates and light source #1 coordinates. Transformation table #1 includes X times Y number of entries (locations), preferably arranged to directly corresponding to the sequential memory addresses of the pixel data values in camera image 51A. When light source #1 and camera 14 each have fixed positions with respect to one another and/or scene 10, transformation table #1 can be precalculated and populated with data using known matrix algebra relationships between the world space coordinate system 20 and the light source #1 and camera 14 perspectives. Thus, the resulting transformation table #1 can be used to quickly determine which, if any, of pixels 60' (light image) correspond to a specific pixel 60 (camera image).

Similarly a second transformation table (#2) 51I is further depicted in Fig. 3. Transformation table #2, is used, in accordance with certain embodiments of the invention to provide for rapid transformation of pixels between camera coordinates and light source #2 coordinates. Transformation

21

table #2 includes X times Y number of entries (locations), preferably arranged to directly corresponding to the sequential memory addresses of the pixel data values in camera image 51A. When light source #2 and camera 14 each have fixed

5   positions, transformation table #2 can be precalculated and populated with data using known matrix algebra relationships between the world space coordinate system 20 and the light source #2 and camera 14 perspectives. Thus, the resulting transformation table #1 can be used to quickly determine

10   which, if any, of pixels 60" (light image) correspond to a specific pixel 60 (camera image).

By using transformation tables 51H-I, and precalculating the data therein, a substantial amount of processing time is saved when compared to calculating the same data "on-the-fly".

15   This additional cost savings can be especially critical for interactive 3D graphic applications designed to operate on slower, lower-end, and/or older computer systems.

Reference is now made to Fig. 4, which depicts an exemplary flow-chart for an improved shadow rendering process

20   100, in accordance with certain embodiments of the present invention. Process 100 can be implemented, for example, in computer system 40 to render lighting and shadows for scene 10. In step 102, the camera's view is rendered using

22

conventional methods, thereby producing camera image 51A and camera depth 51B. In step 104, the view from a light source is rendered. For example, in step 104, rendering the view from light source #1 produces light #1 image 51C and light #1

5   depth 51D.

Next, in step 106, a pixel 60 in camera image 51A is transformed or is otherwise used to determine a corresponding pixel 60' in light #1 image 51C (and light #1 depth 51D). The transformation calculations needed to move between coordinates

10   in step 106 can be precalculated and stored, for example, in transformation table #1 provided that camera 14 and light source #1 are fixed with respect to scene 10. Alternatively, such transformation calculations can be conducted on-the-fly provided adequate processing capability. If either camera 14

15   or light source #1 are not fixed with respect to scene 10, then the transformation calculations can be conducted on-the-fly (i.e., in real-time between frames).

In step 108, if the transformed pixel identified in step 106 is illuminated by the light source, then the corresponding

20   pixel data value in the light image is added to the light accumulation buffer 51G. For example, considering pixel 60', from the Z-value associated with pixel 60' (as identified in light #1 depth 51D) it can be determined if pixel 60' is

23

lighted by, or shaded from, light source #1. If pixel 60' is lighted, then the corresponding (RGB) pixel data value from light #1 image 51C is added to light accumulation buffer 51G. If pixel 60' is not-lighted (i.e., is shaded), then no

5 change is made to light accumulation buffer 51G.

Steps 106 and 108 are repeated for each of the pixels in camera image 51A, as depicted by the iteration provided by step 110. After all of the pixels in camera image 51A have been processed in steps 106-110, then, according to step 112,

10 the process is repeated for each additional light source. Step 112 leads back to step 104, wherein the next light source view is rendered. For example, steps 104-110 can be repeated again for light source #2 (or #N light sources). This iterative process allows for each of the pixel data values

15 within light accumulation buffer 51G to be incrementally increased in a manner that matches the cumulative light cast by the multiple light sources on each pixel.

After each of the light sources has been processed in accord with steps 104 through 112, then in step 114, a pixel

20 data value from camera image 51A is multiplied by a corresponding pixel data value from light accumulation buffer 51G and the results are stored in camera image 51A. Step 114 is repeated for each of the pixels in camera image 51A, as

24

depicted by iterative step 116. Then, in step 118, a resulting camera image 51A is further rendered in accord with other processes, as needed, and eventually stored in frame buffer 50 and displayed as a frame of data.

5   Process 100, therefore, in accordance with certain aspects of the present invention, provides realistic lighting and shadow rendering while also being computationally efficient. Process 100 can be adapted for a plurality of light sources, either fixed or moving, and for changing camera

10 views. By way of example, the following exemplary pseudocode can be implemented in either hardware of software to provide for process 100:

RENDER EACH VIEW (PARTIAL IF THE LIGHT IS STATIONARY)

CLEAR ACCUM BUFFER

15 *FOR EACH LIGHT SOURCE ...*

FOR EACH PIXEL IN CAMERA IMAGE SP_X, SP_Y ...

TRANSFORM EACH SP TO A LP {LIGHT PIXEL} USING EITHER:

TRANSFORM LOOK-UP TABLE,

20 *OR*

MATRIX TRANSFORMATION CALCULATION

IF LP_Z < LIGHT DEPTH (LP_X, LP_Y) THEN

25

$$ACCUM(SP_X, SP_Y) \mathrel{+}= LIGHT\ IMAGE\ (LP_X, LP_Y)$$

FOR EACH PIXEL IN CAMERA IMAGE ...

$$CAMERA\ IMAGE\ (SP_X, SP_Y) \mathrel{*}= ACCUM\ (SP_X, SP_Y)$$

The flow-chart in Fig. 5 depicts an exemplary process 120
5   that can be included in step 106. Process 120 includes step
122, wherein an applicable transformation table is used to
determine the transformation from camera to light source
viewpoints when the camera and light source are fixed with
respect to scene 10. Otherwise, in step 124, if the camera or
10 light source are not fixed then a matrix transformation
algorithm is used.

An exemplary process 130 is depicted in the flow-chart of
Fig. 6. Process 130 can be used to precalculate transform
tables and/or included in step 124 calculate the transform on-
15 the-fly. In step 132, a transformation is made from world
space coordinate system 20 to camera coordinates. Next, in
step 134, a transformation is made from camera coordinates to
light source coordinates. These types of transformation
processes, which are preferably completed using standard
20 matrix algebra, are well known to those skilled in the art
(see, e.g., Foley et al. text). In step 136, the resulting
transformation data (camera to light source) is provided in a

26

translation table per step 138, or returned to the shadow rendering process per step 140.

Figs 7A-D depict actual computer screen shots for an exemplary 3D scene of an interactive real time graphics rendering application, having an improved shadow rendering capability as described above and configured for a PC, in accordance with certain embodiments of the present invention. The application that generated the images depicted in Figs 7A-D is a computer-based interactive adventure game. In the exemplary scenes, the camera and two light sources are each fixed with respect to the scene. The frame in Fig. 7A is a monochrome photocopy of the colored rendered scene. The frame of Fig. 7A is shown inverted in Fig. 7B to further highlight the shadowing rendered in the scene. Fig. 7C is a essentially a "behind the scenes" view of various image buffers that are used to generate the image of Fig. 7A. Fig 7D is the inverted image of Fig. 7C.

Reference will first be made to the image in Fig. 7A, with reference numerals being added to the inverted image of Fig. 7B and/or the fully lighted image of Figs. 7C-D. In Fig. 7A there is depicted a portion of a room 200 having texture mapped walls 202A-D, floor 204, and ceiling 205. One of the walls 202B has a large mirror 203, in which is shown the

27

reflection from another wall 202D having several pictures hanging thereon. Additional 3D objects within the room include torches 206A and 206B mounted on side walls, 202A and 202C, respectively. A man 208 is shown standing on floor 204

5   and in front of the mirrored wall 202B. Light is being reflected onto a portion of the mirrored wall 202B, the floor 204, the man 208, and on the wall 202D having pictures hanging thereon. The man 208 casts a shadow 210 onto a portion of the floor 204 and the mirrored wall 202B.

10   The frame in Fig. 7A, and in particular, the lighting rendered in the scene is demonstrated by the additional images in Fig. 7C. Room 200′ is depicted in full light in Fig. 7C. From this image of room 200′, one can understand the configuration of the modeled 3D world. Refer to the inverted

15   image of Fig. 7D, as required, to identify a specific reference. Fig. 7C also includes four other images that are used to render the lighting and shadows in room 200′.

There are two light sources that are not visible within room 200′. These light sources are used to simulate the light

20   emitted by simulated torches 206A and 206B. These light sources, which are not shown in Fig. 7C, are directional light sources having fixed positions with respect to room 200. Image 220 represents the light depth data for the first light,

28

which has a circular field of view (or aperture) and is positioned to shine light on a portion of floor 204 and mirrored wall 202B. The information depicted in image 220 corresponds to depth, for example, as stored in a Z-buffer

5 associated with this particular light source. Within depth image 220, as depicted in Fig. 7C, the closer a pixel is to the light source, the darker its image. Thus, for example, the man 208 in Fig. 7C is closest to the light source and as such is rendered the darkest shade. Conversely, the interface

10 (corner) of mirrored wall 202B and floor 204 is the farthest away from the light source and as such is rendered the lightest shade. This is just one convention, obviously the convention could be reversed in other systems. Note, that the image 220 in Fig 7D is inverted and thus so to is the

15 convention.

Also associated with the first light source is a light image 224. In this example, light image 224 includes a graduated pattern of intensity that is highest near the center and lowest near the edges. In this example, light image 224

20 is essentially a light filter based on a cosine function. It is intended, however, that the RGB pixel data values in a light image can include any pattern of RGB data, including static and dynamically changing patterns. For example, in

29

accordance with certain embodiments of the present invention, a light image can include a dynamically changing graphical image. This can be used to simulate light that is reflected from changing surfaces, such as, for example, the light

5   reflected from a rippling body of water. By dynamically changing the light image using a recorded or simulated image of such a body of water, the light source will essentially paint the scene with a rippling light. Taken a further step, the light image can include an animation, motion picture or

10   similar video image that is projected, not unlike a movie projector, on to the various surfaces of the scene. Indeed, this can be used to model a movie projector, wherein 3D objects, such as man 208 can walk in front of the projected light image and have portions of the projected image fall on

15   his body.

With regard to man 208 as depicted in depth image 220, the depth image 220 has been further processed in this example to include data relating to the depth of man 208. This can be accomplished, for example, by comparing previous frames and

20   completing new transform calculations for pixels that have changed and that are in the depth image for the light source. Thus, for example, from the previous frame, man 208 may have moved slightly (e.g., in response to inputs from the user).

30

A portion of the pixels are identified as having changed from the previous frame. The portion of the pixels that changed are then transformed and processed to generate new modified light depth data 228. In this manner, only those portions of

5   the scene that change need to be reprocessed.

For the second light source, a depth image 222 and light image 226 are also provided. Depth image 222 demonstrates that the second light source is positioned facing wall 202D and casting light on a portion of floor 204 and wall 202D.

10   Light image 226 is similar to light image 224. Note that depth image 222 does not include man 208 or any other moving 3D object at this time. Thus, depth image 222 need not be updated at this time.

The various images in Fig. 7C are processed in accordance

15   with process 100. A light accumulation buffer and subsequent frame buffer are created by combining data from the image of room 200', and light images 224 and 226, in accordance with the data in depth images 220 and 222. The resulting frame buffer is that of Fig. 7A, which clearly shows that man 208

20   casts a single shadow 210 as a result of the first light. Additionally, the accumulated light on mirrored wall 202B and 202D are shown.

31

In accordance with still further embodiments of the present invention, if camera 14 is required to periodically change position with respect to scene 10, for example, during an interactive simulation, then process 120 is repeated as

5   necessary to recalculate and update the applicable transform tables, or to generate different, additional transform tables, based on the new camera position.   If the camera is continually moving with respect to scene 10, then rather than manipulating and accumulating light based on pixels, process

10   100 is adapted to manipulate groups of pixels and/or polygons. This allows for a more granular approach, in which a light image can be rendered for each polygon, and light accumulation buffers can be used to accumulate light intensities from the various light sources for each polygon, rather than for each

15   screen pixel.   As such, a plurality of light images can be precomputed, one for each of the polygons (or for groups of polygons), for each of the fixed lights.

Although several preferred embodiments of the methods and arrangements of the present invention have been illustrated in

20   the accompanying Drawings and described in the foregoing Detailed Description, it will be understood that the invention is not limited to the embodiment disclosed, but is capable of numerous   rearrangements,   modifications   and   substitutions

32

without departing from the spirit of the invention as set forth and defined by the following claims.

33